Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	1	"6678744".pn.	US-PGPUB; USPAT	OR	ON	2005/09/23 16:30
L2	10	("5361361" | "5457798" | "5463770" | "5475845" | "5603025" | "5872974" | "5983019" | "5983234" | "6003037" | "6336146").PN.	US-PGPUB; USPAT; USOCR	OR	ON	2005/09/23 16:30
L3	1	("6336146").URPN.	USPAT	OR	ON	2005/09/23 16:31
L4	1	wrapper with interface with stable	USPAT	OR	ON	2005/09/23 16:37
L5	2	wrapper with interface with (decoupl$ or stable)	USPAT	OR	ON	2005/09/23 16:32
L6	28	("4736320" | "4809170" | "4905163" | "4974160" | "5195178" | "5206950" | "5212771" | "5233513" | "5237691" | "5249300" | "5261100" | "5287515" | "5361360" | "5367473" | "5471636" | "5487131" | "5493682" | "5519867" | "5524246" | "5535388" | "5542070" | "5544308" | "5583983" | "5587935" | "5651111" | "5699310" | "5802314" | "5854929").PN.	US-PGPUB; USPAT; USOCR	OR	ON	2005/09/23 16:34
L7	1	wrapper with interface with (application or program) with modular$	USPAT	OR	ON	2005/09/23 16:37
L8	1	wrapper with interface with (application or program) with (modular$ or partial$)	USPAT	OR	ON	2005/09/23 16:37
L9	1	wrapper with interface with (application or program) same (modular$ or partial$)	USPAT	OR	ON	2005/09/23 16:38
L10	4	wrapper with interface with (application or program) same (modular$ or partial$)	US-PGPUB; USPAT	OR	ON	2005/09/23 16:38
L11	112	wrapper with (application or program) same (modular$ or partial$)	US-PGPUB; USPAT	OR	ON	2005/09/23 16:57
L13	6	wrapper with (application or program) same interface same (modular$ or partial$)	US-PGPUB; USPAT	OR	ON	2005/09/23 16:40
L14	3	wrapper with (application or program) same object adj oriented same (modular$ or partial$)	US-PGPUB; USPAT	OR	ON	2005/09/23 16:40
L15	3	wrapper same (application or program) same (object adj oriented) same (modular$ or partial$)	US-PGPUB; USPAT	OR	ON	2005/09/23 16:41

L16	1823	719/328.ccls. or 719/315.ccls.	US-PGPUB; USPAT	OR	ON	2005/09/23 16:41
L17	3457	wrapper.ti,ab.	US-PGPUB; USPAT	OR	ON	2005/09/23 16:41
L18	38	16 and 17	US-PGPUB; USPAT	OR	ON	2005/09/23 16:43
L19	6	wrapper with (application or program) same (modular$ or partial$)	EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/09/23 16:42
L20	3	("5727212" | "6345319" | "6671743").PN.	US-PGPUB; USPAT; USOCR	OR	ON	2005/09/23 16:48
L21	112	(wrapper or surrogtate) with (application or program) same (modular$ or partial$)	US-PGPUB; USPAT	OR	ON	2005/09/23 16:58
L22	12	((wrapper or surrogtate) same (application or program) same (modular$ or partial$)).ti,ab,clm.	US-PGPUB; USPAT	OR	ON	2005/09/23 16:59
L23	7021	Moon-B$.in. or Bankler-B$.in. or Ericsson$.as.	US-PGPUB; USPAT	OR	ON	2005/09/23 16:59
L24	5	wrapp$.ti,ab. and 23	US-PGPUB; USPAT	OR	ON	2005/09/23 17:00
L25	1	independent$ and 1	US-PGPUB; USPAT	OR	ON	2005/09/23 17:03
L26	207	application adj wrapper	US-PGPUB; USPAT	OR	ON	2005/09/23 17:04
L27	3	application adj wrapper same modular$	US-PGPUB; USPAT	OR	ON	2005/09/23 17:04
L28	25	application adj wrapper same interface	US-PGPUB; USPAT	OR	ON	2005/09/23 17:04
L29	2	("5528753" | "6026236").PN.	US-PGPUB; USPAT; USOCR	OR	ON	2005/09/23 17:09

Edit an existing query or compose a new query in the Search Query Display.

Select a search number (#) to:

- Add a query to the Search Query Display
- Combine search queries using AND, OR, or NOT
- Delete a search
- Run a search

Fri, 23 Sep 2005, 5:21:34 PM EST

Search Query Display

Recent Search Queries

#1 (application wrapper)<in>pdfdata

#2 (application wrapper)<in>pdfdata

#3 (application wrapper and interface)<in>pdfdata

#4 (application wrapper and modular)<in>pdfdata

Indexed by Inspec

Terms used **application wrapper modular** Found **4,975** of **161,645**

Sort results by relevance Save results to a Binder Try an Advanced Search

Display results expanded form Search Tips Try this search in The ACM Guide

Open results in a new window

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown Relevance scale

1 Classical computational geometry in GeomNet

Gill Barequet, Stina S. Bridgeman, Christian A. Duncan, Michael T. Goodrich, Roberto Tamassia

August 1997 **Proceedings of the thirteenth annual symposium on Computational geometry**

Full text available: pdf(708.93 KB) Additional Information: full citation, references, citings, index terms

Keywords: Internet computing, cooperative computing, geometric computing, visualization

2 Integrating and customizing heterogeneous e-commerce applications

Anat Eyal, Tova Milo

August 2001 **The VLDB Journal — The International Journal on Very Large Data Bases**, Volume 10 Issue 1

Full text available: pdf(286.63 KB) Additional Information: full citation, abstract, citings, index terms

A broad spectrum of electronic commerce applications is currently available on the Web, providing services in almost any area one can think of. As the number and variety of such applications grow, more business opportunities emerge for providing new services based on the integration and customization of existing applications. (Web shopping malls and support for comparative shopping are just a couple of examples.) Unfortunately, the diversity of applications in each specific domain and the dispar ...

Keywords: Application integration, Data integration, Electronic commerce

3 Document adaptation: Supporting virtual documents in just-in-time hypermedia systems

Li Zhang, Michael Bieber, David Millard, Vincent Oria

October 2004 **Proceedings of the 2004 ACM symposium on Document engineering**

Full text available: pdf(707.51 KB) Additional Information: full citation, abstract, references, index terms

Many analytical or computational applications especially legacy systems create documents and display screens in response to user queries "dynamically" or in "real time". These "virtual documents" do not exist in advance and thus hypermedia features must be generated "just in time" - automatically and dynamically. Additionally the hypermedia features may have to cause target documents to be generated or re-generated. This paper

focuses on the specific challenges faced in hypermedia support for ...

Keywords: dynamic hypermedia functionality, dynamic regeneration, integration architecture, just-in-time hypermedia, re-identification, re-location, virtual documents

4 Radix-4 modular multiplication and exponentiation algorithms for the RSA public-key cryptosystem

Jin-Hua Hong, Cheng-Wen Wu

January 2000 **Proceedings of the 2000 conference on Asia South Pacific design automation**

Full text available: pdf(139.43 KB) Additional Information: full citation, references, citings

5 An automata-theoretic approach to modular model checking

Orna Kupferman, Moshe Y. Vardi

January 2000 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 22 Issue 1

Full text available: pdf(458.27 KB) Additional Information: full citation, abstract, references, citings, index terms

In modular verification the specification of a module consists of two part. One part describes the guaranteed behavior of the module. The other part describes the assumed behavior of the system in which the module is interacting. This is called the assume-guarantee paradigm. In this paper we consider assume-guarantee specifications in which the guarantee is specified by branching temporal formulas. We distinguish between two approaches. In the first approac ...

Keywords: automata, modular verification, temporal logic

6 Modular typechecking for hierarchically extensible datatypes and functions

Todd Millstein, Colin Bleckner, Craig Chambers

September 2004 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 26 Issue 5

Full text available: pdf(665.60 KB) Additional Information: full citation, abstract, references, index terms

One promising approach for adding object-oriented (OO) facilities to functional languages like ML is to generalize the existing datatype and function constructs to be hierarchical and extensible, so that datatype variants simulate classes and function cases simulate methods. This approach allows existing datatypes to be easily extended with both new operations and new variants, resolving a longstanding conflict between the functional and OO styles. However, previous designs based on this approac ...

Keywords: Extensible datatypes, extensible functions, modular typechecking

7 A overview of modular smalltalk

Allen Wirfs-Brock, Brian Wilkerson

January 1988 **ACM SIGPLAN Notices , Conference proceedings on Object-oriented programming systems, languages and applications**, Volume 23 Issue 11

Full text available: pdf(1.23 MB) Additional Information: full citation, abstract, references, citings, index terms

This paper introduces the programming language Modular Smalltalk, a descendant of the Smalltalk-80 programming language. Modular Smalltalk was designed to support teams of software engineers developing production application programs that can run independently

of the environment in which they are developed. We first discuss our motivation for designing Modular Smalltalk. Specifically, we examine the properties of Smalltalk-80 that make it inappropriate for our purposes. We then present an o ...

8 Modular typechecking for hierarchically extensible datatypes and functions
Todd Millstein, Colin Bleckner, Craig Chambers
September 2002 **ACM SIGPLAN Notices , Proceedings of the seventh ACM SIGPLAN international conference on Functional programming**, Volume 37 Issue 9
Full text available: pdf(175.68 KB) Additional Information: full citation, abstract, references, citings, index terms

One promising approach for adding object-oriented (OO) facilities to functional languages like ML is to generalize the existing datatype and function constructs to be hierarchical and extensible, so that datatype variants simulate classes and function cases simulate methods. This approach allows existing datatypes to be easily extended with both new operations and new variants, resolving a long-standing conflict between the functional and OO styles. However, previous designs based on this approa ...

Keywords: extensible datatypes, extensible functions, modular typechecking

9 Relaxed MultiJava: balancing extensibility and modular typechecking
Todd Millstein, Mark Reay, Craig Chambers
October 2003 **ACM SIGPLAN Notices , Proceedings of the 18th annual ACM SIGPLAN conference on Object-oriented programing, systems, languages, and applications**, Volume 38 Issue 11
Full text available: pdf(162.17 KB) Additional Information: full citation, abstract, references, citings, index terms

We present the rationale, design, and implementation of Relaxed MultiJava (RMJ), a backward-compatible extension of Java that allows programmers to add new methods to existing classes and to write multimethods. Previous languages supporting these forms of extensibility either restrict their usage to a limited set of programming idioms that can be modularly typechecked (and modularly compiled) or simply forego modular typechecking altogether. In contrast, RMJ supports the new language features in ...

Keywords: class loader, external methods, modular typechecking, multimethods, relaxed MultiJava

10 Disjunction and modular goal-directed proof search
Matthew Stone
July 2005 **ACM Transactions on Computational Logic (TOCL)**, Volume 6 Issue 3
Full text available: pdf(396.28 KB) Additional Information: full citation, abstract, references, index terms

This article explores goal-directed proof search in first-order multimodal logic. I focus on a family of modal logics which offer the expressive power to specify modular goals and local assumptions. A modular goal must be proved from designated assumptions; conversely, a local assumption can only be used to prove a designated goal. Indefinite modal specifications can avoid combinatorial interactions among independent ambiguities by making separate goals modular and corresponding disjunctive alte ...

Keywords: Modal logic, goal-directed proof, indefinite information, locality, logic programming, modularity

11 A modular partitioning approach for asynchronous circuit synthesis

Ruchir Puri, Jun Gu
June 1994 **Proceedings of the 31st annual conference on Design automation**
Full text available: pdf(217.19 KB) Additional Information: full citation, references, citings, index terms

12 Aspect-oriented software development: Aspect-oriented programming and modular reasoning
Gregor Kiczales, Mira Mezini
May 2005 **Proceedings of the 27th international conference on Software engineering**
Full text available: pdf(113.46 KB) Additional Information: full citation, abstract, references, index terms

Aspects cut new interfaces through the primary decomposition of a system. This implies that in the presence of aspects, the complete interface of a module can only be determined once the complete configuration of modules in the system is known. While this may seem anti-modular, it is an inherent property of crosscutting concerns, and using aspect-oriented programming enables modular reasoning in the presence of such concerns.

Keywords: aspect-oriented programming, modular reasoning, modularity

13 Modular termination of context-sensitive rewriting
Bernhard Gramlich, Salvador Lucas
October 2002 **Proceedings of the 4th ACM SIGPLAN international conference on Principles and practice of declarative programming**
Full text available: pdf(590.43 KB) Additional Information: full citation, abstract, references, citings, index terms

Context-sensitive rewriting (CSR) has recently emerged as an interesting and flexible paradigm that provides a bridge between the abstract world of general rewriting and the (more) applied setting of declarative specification and programming languages such as OBJ*, CafeOBJ, ELAN, and Maude. A natural approach to study properties of programs written in these languages is to model them as context-sensitive rewriting systems. Here we are especially interested in proving termination of such systems, ...

Keywords: context-sensitive rewriting, declarative programming, evaluation strategies, modular analysis and construction of programs, modular proofs of termination, program verification

14 A software tool for modular database design
M. A. Casanova, A. L. Furtado, L. Tucherman
May 1991 **ACM Transactions on Database Systems (TODS)**, Volume 16 Issue 2
Full text available: pdf(1.78 MB) Additional Information: full citation, abstract, references, citings, index terms, review

A modularization discipline for database schemas is first described. The dicipline incorporates both a strategy for enforcing integrity constraints and a tactic for organizing large sets of database structures, integrity constraints, and operations. A software tool that helps the development and maintenance of database schemas modularized according to the discipline is then presented. It offers a user-friendly interface that guides the designer through the various stages of the creation of ...

Keywords: abstract data types, consistency preservation, encapsulation, integrity constraints, logical database design, modular design, module constructors

15 Modular arithmetic and finite field theory: A tutorial

E. Horowitz

March 1971 **Proceedings of the second ACM symposium on Symbolic and algebraic manipulation**

Full text available: pdf(569.15 KB) Additional Information: full citation, abstract, references, citings, index terms

The paradigm of algorithm analysis has achieved major pre-eminence in the field of symbolic and algebraic manipulation in the last few years. A major factor in its success has been the use of modular arithmetic. Application of this technique has proved effective in reducing computing times for algorithms covering a wide variety of symbolic mathematical problems. This paper is intended to review the basic theory underlying modular arithmetic. In addition, attention will be paid to certain pr ...

Keywords: Exact multiplication, Finite fields, Modular arithmetic, Symbol manipulation;

16 Incremental modular decomposition

John H. Muller, Jeremy Spinrad

January 1989 **Journal of the ACM (JACM)**, Volume 36 Issue 1

Full text available: pdf(1.55 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Modular decomposition is a form of graph decomposition that has been discovered independently by researchers in graph theory, game theory, network theory, and other areas. This paper reduces the time needed to find the modular decomposition of a graph from &OHgr;(n3) to &Ogr;(n2). Together with a new algorithm for transitive orientation given in [21], this leads to fast new algorithms for a number of problems in gra ...

17 Modular logic programming

Antonio Brogi, Paolo Mancarella, Dino Pedreschi, Franco Turini

July 1994 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 16 Issue 4

Full text available: pdf(2.41 MB) Additional Information: full citation, abstract, references, citings, index terms

Modularity is a key issue in the design of modern programming languages. When designing modular features for declarative languages in general, and for logic programming languages in particular, the challenge lies in avoiding the superimposition of a complex syntactic and semantic structure over the simple structure of the basic language. The modular framework defined here for logic programming consists of a small number of operations over modules which are (meta-) logically defined and sema ...

Keywords: composition operations, declarative semantics, logic programs, metalogic, modularity, program transformation

18 A comprehensive approach for the development of modular software architecture description languages

Eric M. Dashofy, André van der Hoek, Richard N. Taylor

April 2005 **ACM Transactions on Software Engineering and Methodology (TOSEM)**, Volume 14 Issue 2

Full text available: pdf(3.51 MB) Additional Information: full citation, abstract, references, index terms

Research over the past decade has revealed that modeling software architecture at the level of components and connectors is useful in a growing variety of contexts. This has led to the development of a plethora of notations for representing software architectures, each focusing on different aspects of the systems being modeled. In general, these notations

have been developed without regard to reuse or extension. This makes the effort in adapting an existing notation to a new purpose commensurate ...

Keywords: ArchStudio 3, Architecture description languages, XML, xADL 2.0

19 Modular stratification and magic sets for Datalog programs with negation

Kenneth A. Ross
November 1994 **Journal of the ACM (JACM)**, Volume 41 Issue 6

Full text available: pdf(3.60 MB) Additional Information: full citation, abstract, references, citings, index terms, review

A class of "modularly stratified" logic programs is defined. Modular stratification generalizes stratification and local stratification, while allowing programs that are not expressible as stratified programs. For modularly stratified programs, the well-founded semantics coincides with the stable model semantics and makes every ground literal true or false. Modularly stratified programs are weakly stratified, but the converse is false. Unlike some weakly stratified programs, mod ...

Keywords: deductive databases, magic sets, modular stratification, rule rewriting, stratification, well-sounded semantics

20 Progress in modular simulation environments

Charles R. Standridge, James F. Kelly, Thomas Kelley, Jack Walther
November 1996 **Proceedings of the 28th conference on Winter simulation**

Full text available: pdf(688.59 KB) Additional Information: full citation, references, citings

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

CiteSeer Find: application wrapper Documents Citations

Searching for **PHRASE application wrapper**.
Restrict to: Header Title Order by: Expected citations Hubs Usage Date Try: Google (CiteSeer) Google (Web) Yahoo! MSN CSB DBLP
29 documents found. **Order: number of citations.**

A Query Translation Scheme for Rapid Implementation.. - Papakonstantinou.. (1995) (Correct) (49 citations)
Supported .Information Source N Client **Application Wrapper** 1 Wrapper N (b) Figure 1: a) Accessing
www.db.ucsd.edu/publications/querytran.ps

Ozone: Integrating Structured and Semistructured Data - Lahiri, Abiteboul, Widom (Correct) (23 citations)
the full functionality of Ozone as a **wrapper application** on top of the O 2 [BDK92] ODMG-compliant
db.stanford.edu/pub/papers/ozone.ps

Using Declarative Descriptions to Model User.. - Browne, Dávila.. (1997) (Correct) (8 citations)
application interface specification (the **Application Wrapper**) does not specify any semantics, it is not communicating end-user actions to MM 4. The **Application Wrapper** The fourth component that we use is
www.cc.gatech.edu/ftp/groups/mastermind/BookChap97/chapter.ps

A Tool for Securely Integrating Legacy Systems into a.. - Souder, Mancoridis (1999) (Correct) (6 citations)
environment such as the Internet. Thus, the **application wrapper** must not only distribute the application's object model and host security. Through the **wrapper, application** ser- Figure 8. Distributed Dot Tool The
www.mcs.drexel.edu/~smancori/research/papers/wcre99.pdf

The gSOAP Toolkit for Web Services and Peer-To-Peer.. - van Engelen, Gallivan (2002) (Correct) (5 citations)
cost somewhat, but requires the writing of **application wrapper** routines by hand which is costly and error
www.cs.fsu.edu/~engelen/ccgrid.pdf

Using Graph Patterns to Extract Scenarios - Jingwei Wu Ahmed (2002) (Correct) (5 citations)
as Image Viewer, Music View, and Mozilla (a **wrapper application** of the Mozilla web browser)and (5) the
plg.uwaterloo.ca/~aeehassa/home/pubs/iwpc2002.pdf

A Development Environment for Building Component-Based .. - Wiil, Nurnberg.. (2000) (Correct) (4 citations)
services such as frontend services (e.g.an **application wrapper**)middleware services (e.g.a
www.cs.aue.auc.dk/~kock/Publications/Construct/ht00tb.pdf

Reconfigurable Router Modules Using Network Protocol Wrappers - Braun, Lockwood, Waldvogel (2001) (Correct) (3 citations)
one component, which has four interfaces **Application Wrapper** Wrapper Fig. 2. Wrapper concept 4 as a
marcel.wanda.ch/Publications/Obiwan.pdf

Design Issues of Database Access in a CORBA Environment - Leser, Tai, Busse (1998) (Correct) (3 citations)
Common Facilities CORBA Object Legacy **Application Wrapper** Figure 1. Object Management Architecture
salmon.cs.tu-berlin.de:8888/leser/pub_n_pres/ihs98.ps

HEALERS: A Toolkit for Enhancing the Robustness and Security.. - Fetzer, Xiao (2003) (Correct) (1 citation)
user application user application user **application wrapper** profiling Shared Libraries robustness
www.research.att.com/~christof/papers/preprint-DSN2003c.pdf

A Goal-driven Auto-Configuration Tool for the Distributed.. - Jeanine (2000) (Correct) (1 citation)
workflow engines at different sites. **Wrapper Application** Program Workflow Interpreter ComMgr
!History Mgmt Workflow Spec. Worklist Mgmt **Wrapper Application** Program **Wrapper Application** Program Worklist Mgmt **Wrapper Application** Program **Wrapper Application** Program **Wrapper Application** Program
www-dbs.cs.uni-sb.de/~gillmann/Publications/Demo-SIGMOD-submitted.pdf

Active Object-Relational Mediators - Kudrass, Loew, Buchmann (1996) (Correct) (1 citation)
to possible consistency violations. An **application wrapper** surrounds complete legacy systems, both
ftp.dvs1.informatik.tu-darmstadt.de/pub/reports+talks/KuLB96.ps.gz

Sybil: Supporting Heterogeneous Database Interoperability with.. - Roger King (1997) (Correct) (1 citation)
to access a set of data systems a new **application or wrapper** is built) and complete integration (each database systems, a new application or **application wrapper** is built. Clearly, this produces a system
www.cs.colorado.edu/~sanctuary/./Papers/ngits.ps

Interoperability of Parallel Systems: Running PVM Applications in .. - Harper (1995) (Correct) (1 citation)
variables and member functions. In (a) MPL **Wrapper Application** Code Legion Object Interface Protocol
ssrnet.snu.ac.kr/~parkmj/w/cs-95-23.ps

Measuring the Covert Channel Bandwidth - In The Nrl (Correct)
Admin Audit Low Lan Lan High Application Low **Wrapper Application** High Wrapper And Fig. 1. Network Nrl Pump
www.sti.uniurb.it/aldini/NRLPumpDraft.ps

Scientific Applications as Web Services: A Simple Guide - Marlon Pierce Geoffrey (Correct)
concrete, we will develop a simple **wrapper application** for a code, Disloc [2]which is used in
grids.ucs.indiana.edu/ptliupages/publications/cise_WSforeScience.pdf

Towards Hypermedia Support for Database Systems - Anirban Bhaumik Anirbanbhaumik (2001) (Correct)
them. Developers write an independent **application "wrapper"** and a set of mapping rules for each. Note by the Freight Database Wrapper Module. **Application Wrappers**, like user interface wrappers, manage the
www.cis.njit.edu/~bieber/pub/hicss01/hicss01-database.pdf

Application Web Services - Marlon Pierce Choonhan (Correct)
interface is the XML abstraction of this **wrapper application** interface and can be viewed as a list of
grids.ucs.indiana.edu/ptliupages/publications/AWS.pdf

Quality Management in MSIS - Adriana Marotta Universit (Correct)
There are three layers :source, mediation and **application**. The source layer contains each source with according to the requirements coming from the **application** layer. The **application** layer provides the user layer contains each source with its associated **wrapper**, which translates queries and queries' responses
www.fing.edu.uy/inco/pedeciba/bibliote/reptec/TR0303.pdf

Type Inference With Recursive Types At Different Ranks - Pericas-Geertsen (1994) (Correct)
if FIX is defined as an operator because its **application** to a function may require a type beyond that #int)t.list(t) int) for the **application** (FIX L) to type check. Clearly, the type to
cs-people.bu.edu/santiago/Papers/Per:MT-BU-1999.ps

First 20 documents Next 20

Try your query at: Google (CiteSeer) Google (Web) Yahoo! MSN CSB DBLP